

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method of fabricating a reconfigurable processor ~~for running moderately complex programming applications~~, comprising:

- ~~(a) providing source code for a programming application;~~
- ~~(b) entering the source code in a control flow graph-generating compiler to produce producing, from a source code, a control data flow graph of data flow control flow and branch points;~~
- ~~(c) extracting, from the control flow graph, basic code blocks of code lying between branch points;~~
- ~~(d) from the code lying between the branch points generating intermediate data flow graphs from the basic code blocks lying between the branch points;~~
- ~~(e) identifying clusters shared among the intermediate data flow graphs at the highest level of granularity;~~
- ~~(f) from the identified clusters determine determining, from the identified clusters, the an unscheduled largest common subgraph shared among the intermediate data flow graphs;~~
- ~~(g) scheduling the unscheduled largest common subgraph for fast accomplishment of operations in the les;~~
- ~~(h) applying the scheduled les largest common subgraph to the intermediate data flow graphs, including replacing the unscheduled les largest common subgraph therein;~~
- ~~(i) scheduling the intermediate data flow graphs containing the les's largest common subgraph for fast accomplishment of operations in the intermediate data flow graphs to derive data patches having operations and timings of for each individual intermediate data flow graphs;~~

(j) combining the data patches to ~~include combine~~ operations and timing of the ~~lesglargest common subgraph~~ with the operations and timings of ~~for each individual ones of~~ intermediate ~~subgraph data flow graphs~~ that are outside the ~~lesglargest common subgraph~~;

(k) ~~from the combined data patches~~ scheduling for process time reduction, ~~from the combined data patches~~, multiple uses of the ~~lesglargest common subgraph~~ operations and timings ~~necessary usable~~ to accomplish operations and timings of all intermediate ~~subgraph data flow graphs~~ employing the ~~lesglargest common subgraph~~; and

(l) implementing, in hardware having mixed granularities, the operations and timing of the ~~lesglargest common subgraph~~ including:

- (i) partitioning;
- (ii) placing; and
- (iii) interconnection routing.

2. (Currently Amended) ~~In a~~ method of making an integrated circuit ~~for use as a hardware implemented part of a programmed operation implemented in software and hardware,~~ comprising:

~~the improvement comprising~~ identifying hardware circuit elements for execution of a largest common subgraph common among a set of flow graphs representing ~~the a~~ programmed operation;

partitioning into blocks the hardware circuit elements;

arranging the blocks on an area representative of an available area of a surface of a substrate on which the circuit elements are to be formed;

routing interconnections among the blocks;

partitioning into sub-blocks the hardware circuit elements of ~~each individual~~ blocks; and

arranging ~~each individual~~ sub-blocks on ~~an a~~ portion of the area representative of ~~the on~~ which the block from which it ~~has the block has~~ been partitioned, routing interconnections among the sub-blocks, and iteratively partitioning and routing among lesser sub-blocks until the

~~individual hardware~~ circuit elements have been placed and routed.

3. (Currently Amended) The method according to claim 2, wherein ~~the steps of~~said routing interconnections among the blocks includes ~~comprise~~ locating conductors and switches for interconnections among blocks, sub-blocks, and circuit elements.

4. (Currently Amended) The method according to claim 3, wherein said locating conductors and switches ~~further comprises~~ includes locating variable switches to effect variable conductive paths among the blocks, sub-blocks, and circuit elements.

5. (Currently Amended) A method of scheduling process elements of hardware implementing a program operation, comprising:

~~(a)~~ developing a control data flow graph from ~~the~~ software;

~~(b)~~ using a first, non-exhaustive scheduling algorithm to ~~relatively quickly~~ arrive at a first scheduling of the process elements;

~~(c)~~ using a second ~~more exhaustive~~ exhaustive or non-exhaustive scheduling algorithm for at least one and less than all selected paths of the control data flow graph to reduce ~~the a~~ time of execution thereof; and

~~(d) once all paths of the control data flow graph have been scheduled, including all of the second more exhaustive scheduling, merging all of schedules, respecting according to~~ data and resource dependencies.

6. (Currently Amended) The method of scheduling according to claim 5, wherein said using a first non-exhaustive scheduling algorithm ~~step (b) comprises~~ includes PCP-Partial Critical Path scheduling.

7. (Currently Amended) The method of scheduling according to either claim 5 or 6, wherein said using a second exhaustive or non-exhaustive scheduling algorithm ~~step (c) comprises~~ includes branch and bound ~~based~~ bound-based scheduling.

8. (Currently Amended) ~~An dedicated~~ integrated circuit ~~for performing~~configured to perform the program operation ~~having processing elements~~ scheduled according to claim 5.

9. (Currently Amended) ~~An dedicated~~ integrated circuit ~~for performing~~configured to perform the program operation ~~having processing elements~~ scheduled according to claim 6.

10. (Currently Amended) ~~An dedicated~~ integrated circuit ~~for performing~~configured to perform the program operation ~~having processing elements~~ scheduled according to claim 7.

11. (Currently Amended) ~~The A method of forming an application-specific reconfigurable circuit, comprising:~~

~~(a)~~ providing source code for an application to be run by ~~the an application-specific reconfigurable circuit;~~

~~(b)~~ deriving from the source code, flow graphs representing separate portions of the application;

~~(c)~~ identifying at least one largest common flow graph from at least two of the separate portions of the application; and

~~(d)~~ ~~in hardware~~, configuring circuitry of the application-specific reconfigurable circuit to be shared by the separate portions of the application.

12. (Currently Amended) A method of fabricating an integrated circuit implementing multiple program operations, comprising:

~~(a)~~ providing source code for the multiple program operations;

~~(b)~~ deriving control flow graphs for selected multiple program operations of a source code;

~~(c)~~ identifying basic blocks of the control flow graphs;

~~(d)~~ developing data flow graphs of for at least a plurality two or more of the basic blocks;

~~(e)~~identifying a common subgraph shared by at least a pair of the basic blocks of control flow graphs of the separate program operations;

~~(f)~~scheduling that makes up the functions (+, -, *)the common subgraph to quicken shared processes ~~occurrences~~ represented by the common subgraph;

~~(g)~~scheduling the quickened shared processes ~~of the common subgraph~~ for operation in each individual ones of the multiple program operations;

~~(h)~~overall-scheduling of processing units to carry out the common subgraph, by including:

~~(i)~~clustering the shared processes ~~of the processing units of the common subgraph~~ into a macroblock having nodes representing the shared processes of common subgraph and at least a plurality of unconditional, conditional, and reconfiguration edges running between nodes;

~~(ii)~~determining ~~the a~~ relative delay among the possible paths through the common subgraph;

~~(iii)~~performing branch and bound scheduling for ~~at least the longest delay time~~the longest-delay-time path, and for less than all possible paths through the common subgraph including at least a longest-delay-time path; and

~~(iv)~~merging all ~~of the~~ schedules; and

~~(i)~~laying out ~~the an~~ arrangement of circuit elements for implementation of the integrated circuit in hardware, including;

~~(i)~~grouping the circuit elements into first level clusters; and

~~(ii)~~placing the first level clusters by grouping the first level clusters together to form second level clusters and placing the second level clusters.

13. (Currently Amended) The method of ~~fabricating an integrated circuit according to claim 12,~~ wherein ~~step (e)~~said identifying a common subgraph shared by at least a pair of the basic blocks ~~comprises~~ includes identifying seed basic blocks by identifying candidate seed basic blocks among the identified basic blocks of the ~~at least a plurality of~~ control flow graphs, and

comparing candidate seed basic blocks from control flow graphs of separate program operations.

14. (Currently Amended) The method of ~~fabricating an integrated circuit according to~~ claim 13, wherein said identifying seed basic blocks ~~comprises~~ includes identifying ones of the basic blocks that lie inside a loop.

15. (Currently Amended) The method of ~~fabricating an integrated circuit according to~~ claim 14, wherein said identifying ones of the basic blocks that lie inside a loop ~~comprises~~ includes identifying one of:

- (i) ~~a~~ single nested level loop with only one basic block;
- (ii) ~~a~~ single nested level loop with more than one basic block; and
- (iii) ~~a~~ multi-level nested loop.

16. (Currently Amended) The method of ~~fabricating an integrated circuit according to~~ claim 14, wherein said identifying ones of the basic blocks that lie inside a loop ~~comprises~~ includes identifying one of:

- (i) ~~a~~ single nested level loop with more than one basic block; and
- (ii) ~~a~~ multi-level nested loop.

17. (Currently Amended) The method of ~~fabricating an integrated circuit according to~~ claim 16, wherein said identifying seed basic blocks further ~~comprises~~ includes identifying basic blocks of control flow graphs of separate program operations under like control.

18. (Currently Amended) The method of ~~fabricating an integrated circuit according to~~ claim 17, wherein said identifying seed basic blocks further ~~comprises~~ includes determining a count of each individual operation types in a basic block ~~of similar class of decision, merge or pass.~~

19. (Currently Amended) The method of ~~fabricating an integrated circuit according to~~ claim 18, wherein said identifying seed basic blocks further ~~comprises~~ includes examining edges in a data

flow graph of candidate seed basic blocks of control flow graphs from the separate programming operations.

20. (Currently Amended) The method of ~~fabricating an integrated circuit according to~~ claim 19, wherein said examining edges ~~comprises~~ includes classifying edges ~~in the data flow graphs on the bases of~~ based on source and destination node operation type.

21. (Currently Amended) The method of ~~fabricating an integrated circuit according to~~ claim 20, wherein said examining edges includes eliminating edges of one data flow graph having a ~~source operation to destination operation~~ source operation to destination operation not found in the ~~another data flow graph the having edges of which are being examined under examination.~~

22. (Currently Amended) The method of ~~fabricating an integrated circuit according to~~ claim 21, further comprising ~~accomplishing~~ implementing the ~~eliminated edges eliminated in a circuit~~ other than in an application specific integrated circuit (ASIC).

23. (Currently Amended) The method of ~~fabricating an integrated circuit according to~~ claim 22, wherein said ~~accomplishing~~ implementing the ~~eliminated edges eliminated in other than ASIC~~ comprises accomplishing includes implementing the ~~eliminated edges eliminated with in one or more~~ look up tables (LUTs).

24. (Currently Amended) The method of ~~fabricating an integrated circuit according to~~ claim 20, wherein said examining edges further ~~comprises~~ includes comparing associativity among edges being compared.

25. (Currently Amended) The method of ~~fabricating an integrated circuit according to~~ claim 24, wherein said comparing associativity ~~comprises~~ includes determining numbers of predecessor, siblings, companions, and successors of edges being compared.

26. (Currently Amended) The method of ~~fabricating an integrated circuit according to~~ claim 12, wherein ~~step (A)~~ said scheduling the shared processes represented by the common subgraph

includes ~~comprises~~-ASAP scheduling the common subgraph.

27. (Currently Amended) The method of ~~fabricating an integrated circuit according to claim 12 or 23~~, further comprising providing the common operations ~~in of~~ the common subgraph in an application specific integrated circuit ~~(ASIC)~~.

28. (Currently Amended) The method of ~~fabricating an integrated circuit according to claim 12~~, further comprising:

identifying at least one ~~further other~~ common subgraph shared by the at least a pair of the basic blocks and; ~~operating on the at least one further common subgraph pursuant to steps (f) (i)~~

scheduling other shared processes represented by the other common subgraph;

scheduling the other shared processes for operation in individual ones of the multiple program operations; and

laying out another arrangement of other circuit elements for implementation of the integrated circuit in hardware, including;

grouping the other circuit elements into other first level clusters; and

placing the other first level clusters by grouping the other first level clusters together to form other second level clusters and placing the other second level clusters.

29. (Currently Amended) The method of ~~fabricating an integrated circuit according to claim 12~~, wherein ~~step (e)~~ said identifying a common subgraph shared by at least a pair of the basic blocks includes comprises-identifying the a largest common subgraph shared by the at least a pair of the basic blocks.

30. (Currently Amended) The method of ~~fabricating an integrated circuit according to claim 12~~, wherein ~~step (f)~~ said scheduling the shared processes represented by the common subgraph includes comprises-providing switching of differing delays among processes of the common subgraph to effect the subgraphs operating each individual ones of the selected multiple program operations.

31. (Currently Amended) The method of ~~fabricating an integrated circuit according to claim 30,~~
wherein said providing switching ~~comprises~~ includes providing multiplexers operative to ~~switch~~
apply alternative delays between processes of the common subgraph.

32. (Currently Amended) An integrated circuit fabricated by the method of claim 12.

33. (Currently Amended) ~~Computer programming having routines implementing~~ A computer-
readable medium comprising stored programming instructions which, in response to execution
by a processor of an apparatus, causes the apparatus to perform the method of claim 12.

34. (Currently Amended) ~~In a~~ method of fabricating a reconfigurable integrated circuit,
comprising ~~including developing a data flow graph for at least a portion of the operations of the~~
integrated circuit;

the improvement comprising:

(a) ~~scheduling the at least a portion of the operations by calculating the a delay along~~
each individual paths through the a data flow graph, wherein the paths go from a processing
element being scheduled to a sink node of the data flow graph, wherein said scheduling
including ~~includes:~~

(i) ~~adding to edges of the data flow graph reconfiguration edges representing a~~
reconfiguration of that part of the integrated circuit effecting the at least a portion of the
operations; and

(ii) ~~including in the calculation of delay along each path the an effect of the~~
reconfiguration on a delay time; and

(b) ~~scheduling first, as the a longest in duration path, the a plurality of longest of~~
processing times of processing elements, including the delay and the reconfiguration delay and
the calculated delay of step (a).

35. (Currently Amended) ~~A~~ The method of ~~fabricating a reconfigurable integrated circuit~~

~~according to claim 34, further comprising scheduling all one or more shorter in duration shorter-in-duration paths within the a time established for the longest in duration longest-in-duration path.~~

36. (Currently Amended) An integrated circuit fabricated ~~by according to~~ the method of claim 34.

37. (Currently Amended) A computer-readable medium comprising stored programming instructions for execution by a computing device, the instructions comprising: Computer programming having routines implementing the method of claim 34.

instructions to schedule at least a portion of operations by calculating a delay along individual paths through a data flow graph, wherein the individual paths go from a processing element being scheduled to a sink node of the data flow graph;

instructions to add to edges of the data flow graph reconfiguration edges representing a reconfiguration of that part of the integrated circuit effecting the at least a portion of the operations; and

instructions to include an effect of the reconfiguration on a delay time; and

instructions to schedule, as a longest in duration path, a plurality of longest of processing times of processing elements, including the delay and the reconfiguration delay.

38. (Currently Amended) ~~In a~~ A method of fabricating a reconfigurable integrated circuit, comprising including developing a data flow graph for at least a portion of the operations of the integrated circuit;

the improvement comprising:

(a) scheduling the at least a portion of the operations of an integrated circuit by including calculating performing a first non-exhaustive calculation of the a plurality of delays along each individual ones of a plurality of partial critical paths through the a data flow graph, wherein the plurality of delays are from a processing element being scheduled to a sink node of the data flow

graph, and wherein said performing a first calculation results at least in a first longest-in-duration path calculation of a first longest-in-duration path of the plurality of partial critical paths;

(b) calculating with performing a second, non-exhaustive or more-exhaustive calculation the of a delay through the path determined to be the longest-in-duration first longest-in-duration path in step (a), wherein said performing a second non-exhaustive or exhaustive calculation results in a second longest-in-duration path calculation; and

(c) determining whether the second longest-in-duration path calculation of step (b) confirms the first longest-in-duration path calculation of the longest-in-duration path calculation of step (a); and

performing one of:

(i) a scheduling of the first the longest-in-duration path calculation in step (a) if step (b) confirms step (a) upon a determination that the second longest-in-duration path calculation confirms the first longest-in-duration path calculation; or

(ii) a determining, from among the remaining paths of the plurality of partial critical paths, the a new longest-in-duration path other than the path so determined in step (a) and scheduling first the new longest-in-duration path from among the remaining paths.

39. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to claim 38, wherein step (c) (ii) said determining, from among remaining paths of the plurality of partial critical paths, a new longest-in-duration path includes~~ comprises ~~calculating performing a third non-exhaustive or exhaustive calculation with the second, more exhaustive calculation the of a delay through the a new longest-in-duration path chosen from among the remaining plurality of partial critical paths excluding the first longest-in-duration path.~~

40. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to claim 38 or 39, wherein the more exhaustives~~ second non-exhaustive or exhaustive calculation comprises includes branch and bound calculation.

41. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to claim 38 or 39~~, further comprising scheduling all ~~shorter-in-duration~~ shorter-in-duration paths within ~~the a~~ time established for the ~~longest-in-duration~~ first longest-in-duration path ~~or the new longest-in-duration path~~.

42. (Currently Amended) An integrated circuit fabricated ~~by~~ according to the method of claim 38.

43. (Currently Amended) A computer-readable medium comprising stored programming instructions which, in response to execution by a processor of an apparatus, causes the apparatus to perform ~~A computer programming having routines implementing~~ the method of claim 38.

44. (Currently Amended) ~~In a~~ method of fabricating a reconfigurable integrated circuit, ~~comprising: including developing a data flow graph for at least a portion of the operations of the integrated circuit;~~

~~the improvement comprising:~~

(a) ~~scheduling an operation of a processing element in a loop;~~

(b) ~~scheduling one or more buffer times following the operation of the processing element in a loop;~~

(c) ~~scheduling an operation of all further~~ additional processing elements, ~~wherein the additional processing elements are dependent on the processing element in a loop, and wherein said scheduling an operation schedules the additional processing elements to begin beginning after the one or more buffer times to permit communication, to the further, dependent~~ additional processing elements, of the status of the additional processing elements ~~in a loop~~.

45. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to claim 44~~, wherein ~~step (a)~~ said scheduling an operation of a processing element in a loop ~~includes~~ comprises scheduling an estimated ~~most probably~~ number of iterations of the operation of the processing element ~~in a loop~~.

46. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to claim 44, wherein step (a) said scheduling an operation of a processing element in a loop includes comprises~~ scheduling a single iteration of the operation of the processing element in a loop.

47. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to claim 44, further comprising providing for notification of delay of all of the dependent additional processing elements during the buffer times upon in response to the processing element in the loop~~ iteratively operating in excess of the scheduled operation of step (a).

48. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to claim 47, further comprising providing for delay of a network schedule manager and of a logic schedule manager upon the processing element in the loop~~ iteratively operating in excess of the scheduled operation of step (a).

49. (Currently Amended) An integrated circuit fabricated ~~by~~ according to the method of claim 44.

50. (Currently Amended) A computer-readable medium comprising stored programming instructions which, in response to execution by a processor of an apparatus, causes the apparatus to perform Computer programming having routines implementing the method of claim 44.

51. (Currently Amended) ~~In a~~ A method of fabricating a reconfigurable integrated circuit, comprising: including developing a data flow graph for at least a portion of the operations of the integrated circuit;

the improvement comprising:

(a) providing a first loop having a first processing element;

(b) providing an output of the first loop to a second loop having a second processing element;

(e) providing an output of the second loop to an input of the first loop;

(d) scheduling an identical number of operations of the first processing element and the second processing ~~processive~~ elements; and

(e) scheduling an operation of ~~all further~~ additional processing elements, wherein the additional processing elements are dependent on the first loop and the second loops, and wherein said scheduling an operation schedules the additional processing elements to begin beginning after a buffer time to permit communication of the status of the first loop and the second loops to the ~~further, dependent~~ additional processing elements.

52. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to claim 51, wherein step (a) said providing a first loop having a first processing element includes comprises scheduling an estimated most probably~~ number of iterations of the operation of the first loop and the second loops.

53. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to claim 51, wherein step (a) said providing a first loop having a first processing element includes comprises scheduling a single iteration of the operation of the first loop and the second loops.~~

54. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to claim 51, further comprising providing for notification of delay to all dependent additional~~ processing elements during buffer times when upon the first loop and the second loops iteratively operate operating in excess of the scheduled operation of ~~step (a).~~

55. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to claim 54, further comprising providing for delay of a network schedule manager and a delay of a logic schedule manager when upon the first loop and the second loops iteratively operate operating~~ in excess of the scheduled operation.

56. (Currently Amended) ~~In a~~ A method of fabricating a reconfigurable integrated circuit,

~~comprising; including developing a data flow graph for at least a portion of the operations of the integrated circuit;~~

~~the improvement comprising:~~

~~(a) providing a loop having control nodes within the loop;~~

~~(b) scheduling the loop by:~~

~~(i) scheduling the a longest-in-duration longest-in-duration path through the loop; and~~

~~(e) scheduling an operation of all further processing elements dependent on the loop after a buffer time, wherein said scheduling an operation ~~to~~ permits communication of ~~the a~~ status of the loop to the further, dependent processing elements.~~

57. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to~~ claim 56, wherein ~~step (a)~~ said providing a loop having control nodes within the loop includes ~~comprises~~ scheduling an estimated ~~most probable~~ number of iterations of the operation of the loop, ~~each wherein the iterations~~ employing the longest-in-duration path through the loop.

58. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to~~ claim 56, wherein ~~step (a)~~ said providing a loop having control nodes within the loop includes ~~comprises~~ scheduling a single iteration of the loop employing the longest-in-duration path through the loop.

59. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to~~ claim 56, further comprising providing for notification of delay to all ~~dependent~~ elements during buffer times ~~when upon~~ the loop iteratively ~~operates~~ operating in excess of the scheduled operation ~~of step (a)~~.

60. (Currently Amended) The method of ~~fabricating a reconfigurable integrated circuit according to~~ claim 59, further comprising providing for delay of a network schedule manager and ~~of~~ a logic schedule manager ~~when upon~~ the loop iteratively ~~operates~~ operating in excess of the scheduled operation ~~of step (a)~~.

61. (Original) An integrated circuit fabricated by the method of claim 56.

62. (Currently Amended) A computer-readable medium comprising stored programming instructions for execution by a computing device, the instructions comprising: Computer programming having routines implementing the method of claim 56.

instructions to provide a loop having control nodes within the loop;

instructions to a longest-in-duration path through the loop; and

instructions to schedule an operation of processing elements dependent on the loop after a buffer time, wherein communication of a status of the loop to the processing elements is permitted.